Technical Memorandum No. 33-119

Pressure Distribution in a Hydrostatic Bearing of Multi-Wells

Sandra M. Barondess

Horace P. Phillips

Senol Utku

GPO PRICE $ ______

CFSTI PRICE(S) $ ______

Hard copy (HC) 2.00

Microfiche (MF) .50

ff 653 July 65

JET PROPULSION LABORATORY

CALIFORNIA INSTITUTE OF TECHNOLOGY

PASADENA, CALIFORNIA

March 1, 1963

N66 26149

(ACCESSION NUMBER) (THRU) 1

28 (PAGES) (CODE) 15

CR-74884 (NASA CR OR TMX OR AD NUMBER) (CATEGORY)

FACILITY FORM 602

Technical Memorandum No. 33-119

Pressure Distribution in a Hydrostatic Bearing of Multi-Wells

Sandra M. Barondess
Horace P. Phillips
Senol Utku

R. Stevens, Chief
Research Section

PASADENA, CALIFORNIA
CALIFORNIA INSTITUTE OF TECHNOLOGY
JET PROPULSION LABORATORY

March 1, 1963

Copyright © 1963
Jet Propulsion Laboratory
California Institute of Technology

Prepared Under Contract No. NAS 7-100
National Aeronautics & Space Administration

CONTENTS

FIGURES

ABSTRACT

26149

The Navier-Stokes equations are first specialized for very slow motion to obtain the equation governing the pressure distribution in the hydrostatic bearing. The resulting differential operator of the governing equation is the Laplacian. The solution domain is rectangular and contains two or three pairs of symmetrically located wells of rectangular shape. Any one of the wells may be considered to have unit pressure while the remaining wells and the outside boundaries of the pad have zero pressures. Since the system is completely linear, the pressure distribution on the pad with arbitrary well pressure combinations can be obtained by superposition.

The differential equation is solved numerically with the aid of a digital computer. The differential equation is converted to a set of linear simultaneous equations through a finite difference scheme. The coefficient matrix is tridiagonalized through suitable partitioning. Because of the quasi-tridiagonal property of this matrix, the solution is obtained by an upper-lower procedure. Since this procedure causes considerable round-off errors when applied to large sensitive systems, an optional iteration with single steps is provided. A machine object time study with reference to the grid size and number of iterations is included. The effect of truncation errors is demonstrated by production runs of a typical case with different grid sizes.

I. INTRODUCTION

This work was initiated to study the location of the pressure resultants in the hydrostatic bearings of the Advanced Antenna System for the NASA/JPL Deep Space Instrumentation Facility. The knowledge of the pressure distribution on the pad under different well pressure conditions is essential for an efficient and safe hydrostatic bearing design.

The wells and the pad of the hydrostatic bearing are assumed to be rectangular as shown in Fig. 1. The number of well pairs might be two or three. These distances, *a, b, c, d, e,* and *f*, are the geometrical parameters of the problem. The solution is given for the cases in which only one of the wells has unit pressure and all the remaining wells and the exterior boundaries of the pad have zero pressures. Since the differential equation and the boundary conditions are linear, the pressure distribution on the pad for any arbitrary well pressures can be obtained by superposition, as follows.

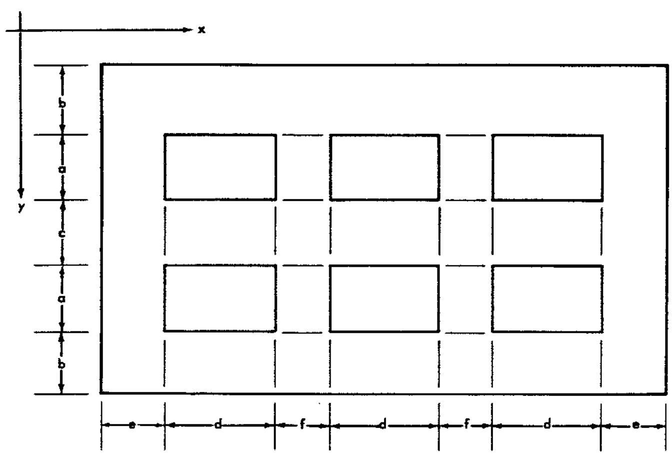

Fig. 1. Geometric parameters of the hydrostatic bearing pad and the wells

Let $p_j(x, y)$, P_j, x_j, and y_j be, respectively, the pressure distribution in the pad, the total thrust, the x and y coordinates of the point of action of the total thrust when the jth well has unit pressure. If α_j is the actual pressure in the jth well, then it follows that

$$p(x,y) = \sum_{j=1}^{N} \alpha_j \, p_j(x,y) \tag{1}$$

$$P = \sum_{j=1}^{N} \alpha_j \, P_j \tag{2}$$

$$X = \frac{\sum_{j=1}^{N} x_j \, P_j \, \alpha_j}{P} \tag{3}$$

$$Y = \frac{\sum_{j=1}^{N} y_j \, P_j \, \alpha_j}{P} \tag{4}$$

where $p(x, y)$, P, X, and Y are, respectively, the actual pressure distribution in the pad, the actual total thrust, the actual x and y coordinates of the point of action of the total thrust, and N is the total number of wells. A general purpose digital computer program is developed to give $p_j\ (x, y)$, P_j, x_j, and y_j for any specified values of the geometric parameters.

II. FORMULATION OF THE PROBLEM

The general motion of a Newtonian fluid can be described by the Navier-Stokes equations (Ref. 1) which are

$$\rho\frac{Du}{Dt} = \overline{X} - \frac{\partial p}{\partial x} + \frac{\partial}{\partial x}\left[\mu\left(2\frac{\partial u}{\partial x} - \frac{2}{3}\operatorname{div}\mathbf{w}\right)\right] + \frac{\partial}{\partial y}\left[\mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)\right] + \frac{\partial}{\partial z}\left[\mu\left(\frac{\partial w}{\partial x} + \frac{\partial u}{\partial z}\right)\right] \quad (5)$$

$$\rho\frac{Dv}{Dt} = \overline{Y} - \frac{\partial p}{\partial y} + \frac{\partial}{\partial y}\left[\mu\left(2\frac{\partial v}{\partial y} - \frac{2}{3}\operatorname{div}\mathbf{w}\right)\right] + \frac{\partial}{\partial z}\left[\mu\left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}\right)\right] + \frac{\partial}{\partial x}\left[\mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)\right] \quad (6)$$

$$\rho\frac{Dw}{Dt} = \overline{Z} - \frac{\partial p}{\partial z} + \frac{\partial}{\partial z}\left[\mu\left(2\frac{\partial w}{\partial z} - \frac{2}{3}\operatorname{div}\mathbf{w}\right)\right] + \frac{\partial}{\partial x}\left[\mu\left(\frac{\partial w}{\partial x} + \frac{\partial u}{\partial z}\right)\right] + \frac{\partial}{\partial y}\left[\mu\left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}\right)\right] \quad (7)$$

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho u)}{\partial x} + \frac{\partial(\rho v)}{\partial y} + \frac{\partial(\rho w)}{\partial z} = 0 \quad (8)$$

where the first three equations are the dynamic equilibrium equations, and the fourth one is the continuity equation. For very slow motion in the pad, the inertial and body forces in the equilibrium equations can be ignored. This yields

$$\operatorname{grad} p = \mu\,\nabla^2\,\mathbf{w} \quad (9)$$

Assuming the fluid is incompressible, the continuity equation can be reduced to

$$\operatorname{div}\mathbf{w} = 0 \quad (10)$$

In scalar form, Eqs. (9 and 10) are

$$\frac{\partial p}{\partial x} = \mu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}\right) \quad (11a)$$

$$\frac{\partial p}{\partial y} = \mu\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial z^2}\right) \quad (11b)$$

$$\frac{\partial p}{\partial z} = \mu\left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2}\right) \quad (11c)$$

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \quad (11d)$$

From Eqs. (9 and 10) one can eliminate w by first taking the divergence of both sides of Eq. (9) and then substituting Eq. (10) into it, as follows.

$$\operatorname{div}\operatorname{grad} p = \operatorname{div}\left[\mu\,\nabla^2\mathbf{w}\right] \quad (12)$$

$$\operatorname{div}\operatorname{grad} p = \mu\,\nabla^2\left[\operatorname{div}\mathbf{w}\right] \quad (13)$$

$$\operatorname{div}\operatorname{grad} p = 0 \quad (14)$$

Eq. (14) is identical with

$$\nabla^2 p = 0 \quad (15)$$

The problem reduces to the solution of the Laplacian in the solution domain shown in Fig. 1 with the prescribed boundary conditions. Since there is no flow in the z direction

$$\frac{\partial p}{\partial z} = 0 \quad (16)$$

Then

$$p = p(x, y) \quad (17)$$

and the Laplacian operator in the chosen coordinate system is

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \quad (18)$$

III. FORMULATION OF NUMERICAL SOLUTION

Equation (15) is solved by means of finite differences. In Fig. 2, the finite difference grid is shown. Assuming an $0(h^2)$ approximation, the difference operator corresponding to the Laplacian is as shown in Fig. 3. In this figure, the unknown mesh function is shown with the symbol $P_{i,j}$ where i is the row number and j the column number in the grid. The application of the difference operator on the unknown mesh function $P_{i,j}$ yields a set of simultaneous linear equations in the form

$$[Q]\{P\} = \{C\} \tag{19}$$

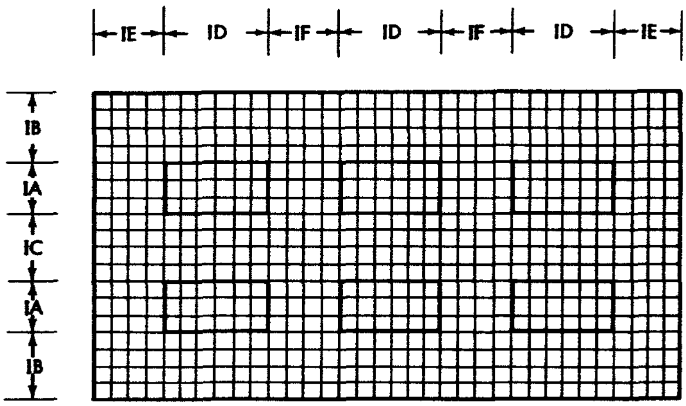

Fig. 2. The finite difference grid

In any one row of the $[Q]$ matrix, there are only five non-zero entries. Relabeling the $P_{i,j}$ terms with a single subscript counting column-wise in the grid, the $[Q]$ matrix will assume the form shown in Fig. 4. Since the mesh function is known on the wells and on the boundaries of the pad, this information can suitably be imposed on $[Q]$ as follows: (1) For the zero values of the $P_{i,j}$ on the exterior boundaries of the pad, the corresponding rows and columns of the augmented matrix are deleted; (2) For the values of the $P_{i,j}$ on the wells, the non-diagonal and diagonal entries of the corresponding rows of the augmented matrix are made respectively zero and one, and the entries of these rows corresponding to $\{C\}$ are made either one or zero depending upon whether the well carries unit or zero pressure, respectively. After this modification $[Q]$ is still a five diagonal matrix as shown in Fig. 4, which can be partitioned as in Fig. 5 to yield a quasi-tridiagonal matrix as in Fig. 6. The entries of the quasi-tridiagonal matrix are square submatrices of order two less than the number of rows in the grid, and the order of the tridiagonal partitioned matrix is two less than the number of columns in the grid. The equations shown in Fig. 6 are of the type of Eq. (19). An upper-lower procedure as summarized below can be used for solution (Ref. 2).

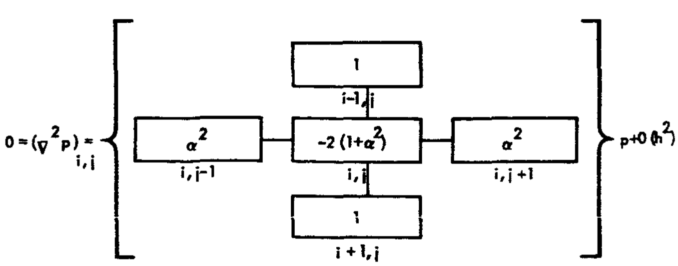

Fig. 3. The finite difference operator associated with ∇^2

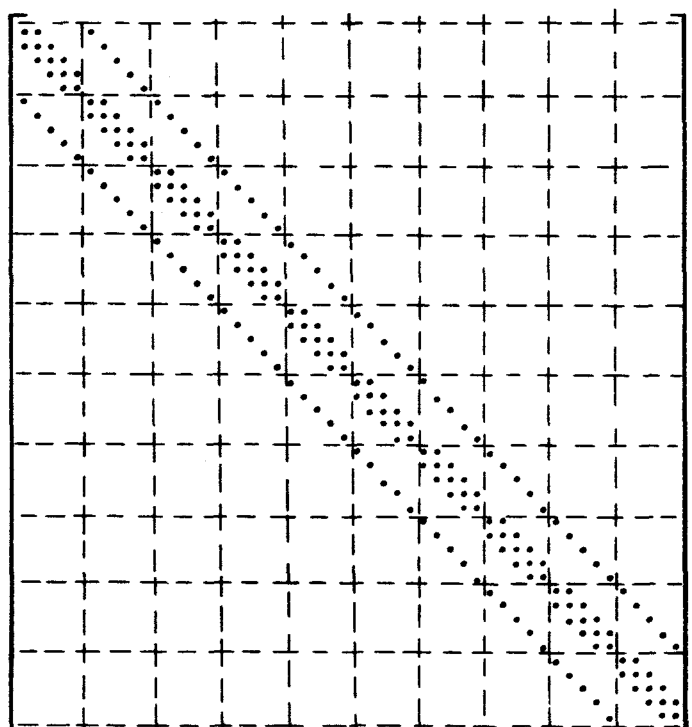

Fig. 4. The coefficient matrix [Q]

Any positive tridiagonal $[Q]$ can be written as

$$[Q] = [L][U] \tag{20}$$

where $[L]$ and $[U]$ are as shown in Fig. 7. The $[R]$ and $[S]$ submatrices of $[L]$ and $[U]$, respectively, can be expressed in terms of the $[A]$ and $[B]$ submatrices of $[Q]$, yielding the following recursive formulas:

$$[S_1] = [B_1] \tag{21}$$

$$[R_n] = [S_{n-1}]^{-1}\,[A_n] \tag{22}$$

$$[S_n] = [B_n] - [S_{n-1}]^{-1}\,[A_n][A_{n-1}] \tag{23}$$

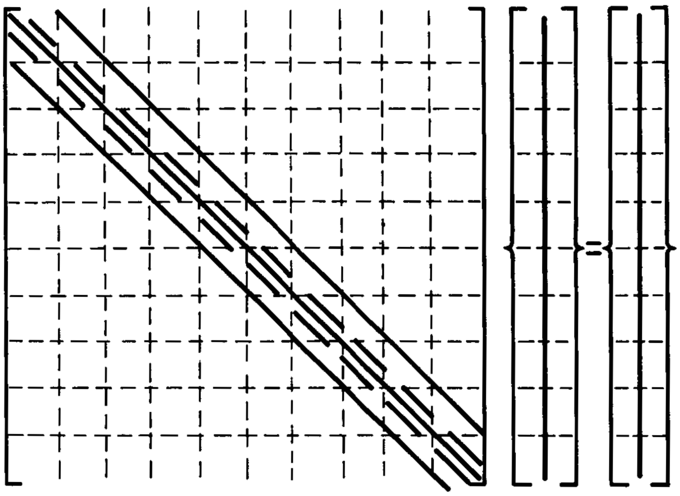

Fig. 5. The partitioning of the set of equations to obtain a quasi-tridiagonal coefficient matrix

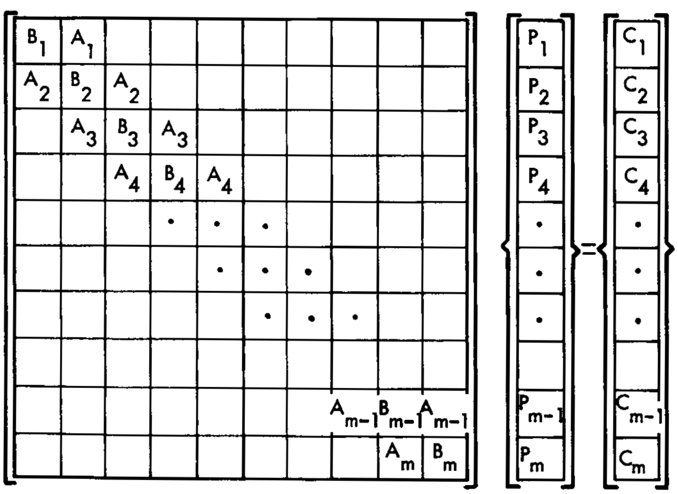

Fig. 6. The set of equations with quasi-tridiagonal coefficient matrix

Having defined $[L]$ and $[U]$, Eq. (19) can be written

$$[L]\,\{Y\} = \{C\} \tag{24}$$

where

$$\{Y\} = [U]\,\{P\} \tag{25}$$

By means of a forward sweep, one can obtain the $\{Y_n\}$ from the formulas

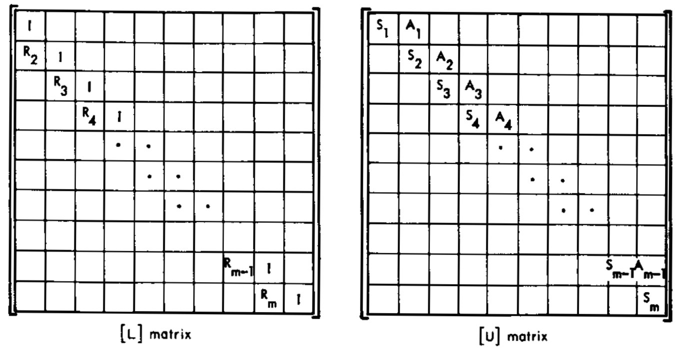

Fig. 7. The [L] and [U] matrices associated with [Q]

$$\{Y_1\} = \{C_1\} \tag{26}$$

$$\{Y_n\} = \{C_n\} - [R_n]\,\{Y_{n-1}\} \tag{27}$$

Having computed the $\{Y_n\}$, the $\{P_n\}$ are computed by a backward sweep from

$$\{P_n\} = [S_n]^{-1}\,\{Y_n\} \tag{28}$$

$$\{P_{n-1}\} = [S_{n-1}]^{-1}\Big\{\{Y_{n-1}\} - [A_{n-1}]\,\{P_n\}\Big\} \tag{29}$$

When the number of unknowns $P_{i,j}$ is of the order of hundreds, the above upper-lower procedure would yield rather large round-off errors in sensitive systems. To improve the results obtained through this procedure, an iteration with single steps is applied (Ref. 3) using the above results as the initial estimate as described below.

By applying the finite difference operator illustrated in Fig. 3 on the mesh function $P_{i,j}$, one writes

$$P_{i,j} = \frac{P_{i-1,j} + P_{i+1,j} + \alpha^2\,(P_{i,j-1} + P_{i,j+1})}{2\,(1 + \alpha^2)} \tag{30}$$

Equation (30) would be satisfied if the $P_{i,j}$'s were the true solutions of Eq. (19). For other cases, Eq. (30) can be rewritten as

$$P^{(s+1)}_{i,j} = \frac{P^{(s)}_{i-1,j} + P^{(s)}_{i+1,j} + \alpha^2\left(P^{(s)}_{i,j-1} + P^{(s)}_{i,j+1}\right)}{2\,(1 + \alpha^2)} \tag{31}$$

for an iteration scheme. In this equation, the superscript in parentheses indicates the number of iterations performed. The correction shown in Eq. (31) should be applied only to those grid points whose pressures are not known. The algorithm of the iteration is such that the grid points are swept in row (or column) sequence, always using the most recently obtained $P_{i,j}$'s.

A Fortran program was developed to perform the above procedures. The flow chart of the program is given in Fig. 8.

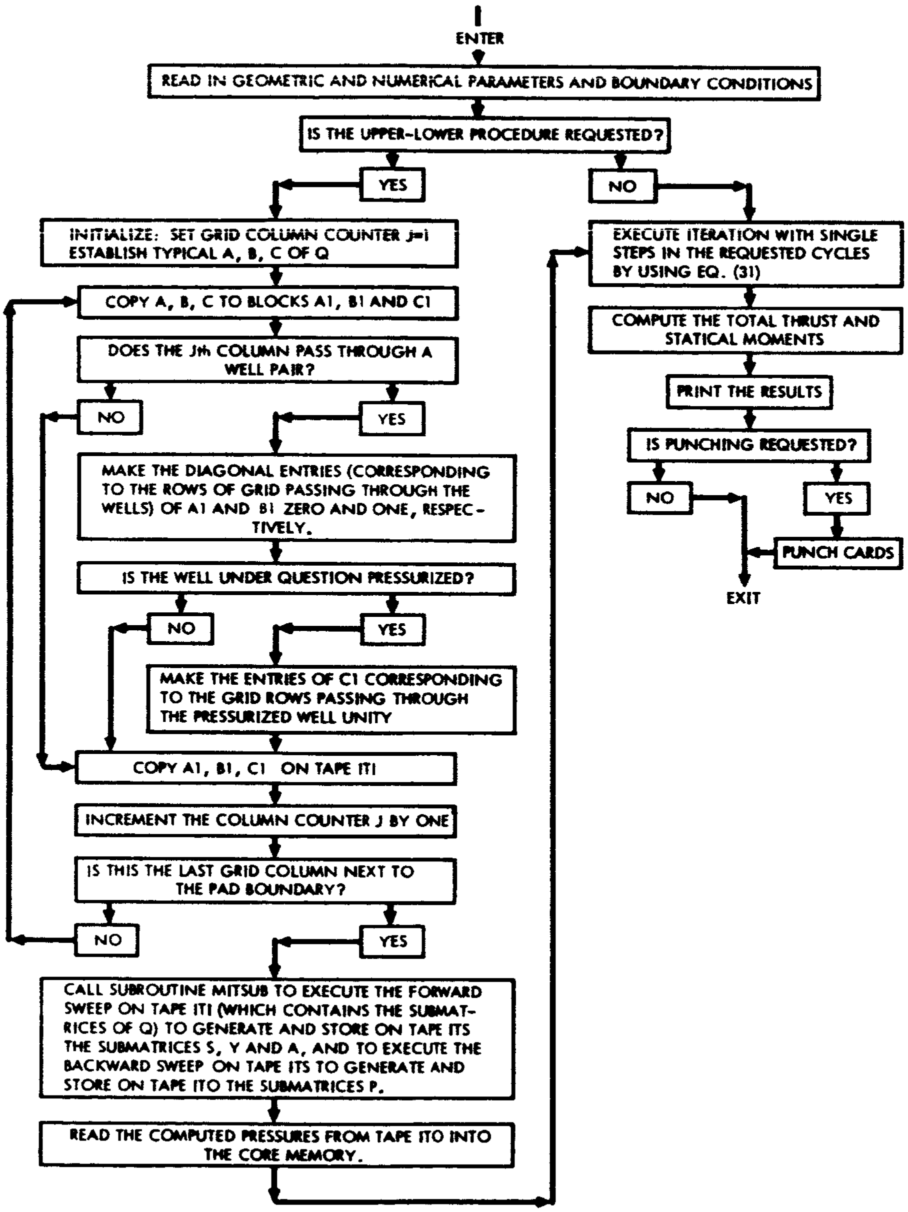

Fig. 8. The flow chart of the program

IV. RESULTS

In Fig. 9, a typical hydrostatic bearing used for the error and object time study is shown. In Fig. 10, thrust versus number of grid points is plotted, where the number of iteration sweeps is taken as a parameter. In Fig. 11, the object time versus number of grid points is plotted, where the number of iteration sweeps is taken as a parameter.

Several test runs indicated that the upper-lower procedure yields considerable round-off errors when the number of unknowns is greater than a few hundred. This is because of the sensitive character of the algebraic system. Therefore it is the authors' suggestion that the program should be run by by-passing the upper-lower procedure, in which case no tapes are necessary. The results given in Fig. 10 and 11 were obtained through iteration only.

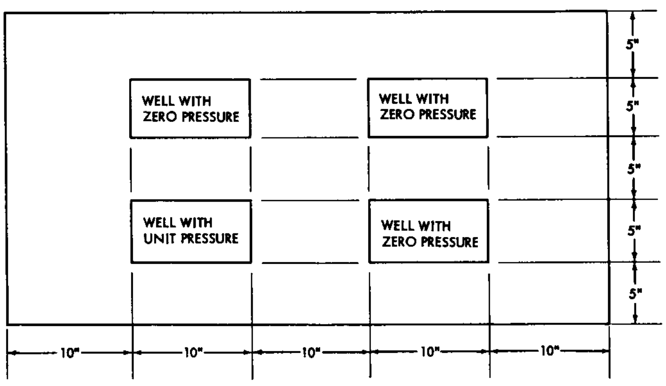

Fig. 9. Hydrostatic bearing configuration used in error and object time studies

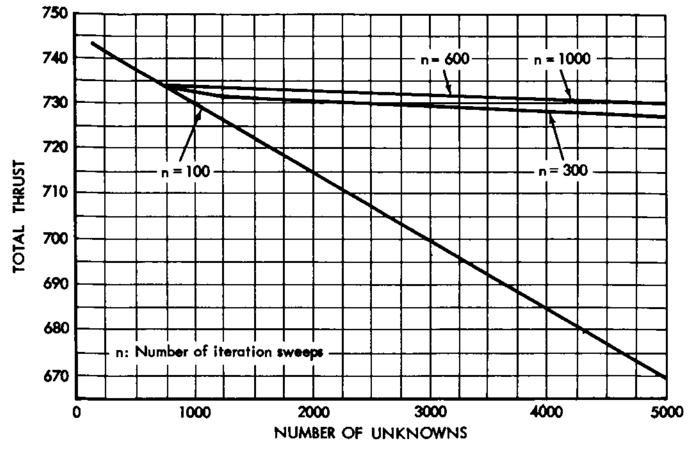

Fig. 10. Total thrust versus number of unknowns for the case shown in Fig. 9 (iterations only)

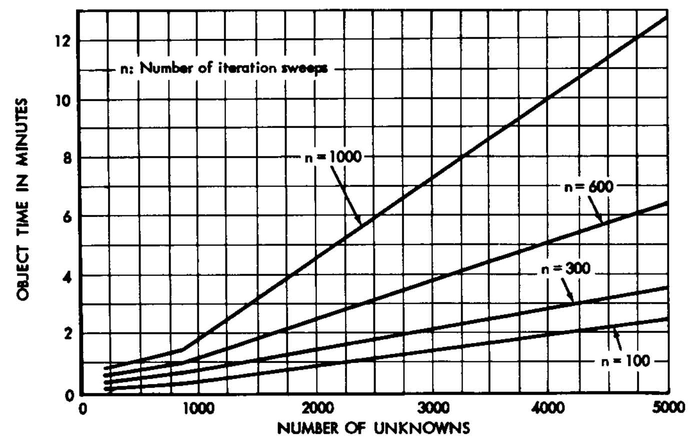

Fig. 11. The object time versus number of unknowns for the case shown in Fig. 9 (iterations only)

NOMENCLATURE

$[A]$, $[B]$, $\{C\}$	Submatrices of the augmented matrix
a, b, c, d, e, f	Geometrical parameters
$\frac{D}{Dt}$	Total derivative with respect to time
div $\mathbf{w}$	$\nabla \cdot \mathbf{w}$
grad p	∇p
$[I]$	Identity matrix
$\mathbf{i}, \mathbf{j}, \mathbf{k}$	Unit vectors of the cartesian coordinate system
$[L]$, $[U]$	Lower and upper matrices of $[Q]$
N	Number of wells in the pad
$p, p(x, y), p_j(x, y)$	Pressure, actual pressure function, and pressure function caused by jth well alone
P, P_j	Actual total thrust and total thrust caused when jth well is pressurized
$[Q]$	Coefficient matrix
$[R]$, $[S]$	Submatrices of $[L]$ and $[U]$
u, v, w	Components of velocity vector
$\mathbf{w}$	Velocity vector
x, y	Cartesian coordinates
x_j, y_j	Coordinates of the point of action of P_j
X, Y	Coordinates of the point of action of actual thrust
$\bar{X}, \bar{Y}, \bar{Z}$	Components of body force
$\{Y\}$	Auxiliary solution vector
α	$\Delta y/\Delta x$, mesh size ratio
α_j	Actual pressure in the jth well
$\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z}, \frac{\partial}{\partial t}$	Partial derivative operators
ρ	Mass per unit volume
μ	Viscosity
∇	Del operator
∇^2	Laplacian operator

REFERENCES

1. Schlichting, Hermann, *Boundary Layer Theory,* McGraw-Hill Book Company, Inc., 1955.
2. Schechter, S., "Quasi-tridiagonal Matrices and Type-insensitive Equations," New York University, *Applied Mathematics Center Report TID-4500,* May 1959.
3. Crandall, Stephen H., *Engineering Analysis,* McGraw-Hill Book Company, Inc., 1956.

APPENDIX

The program, written in Fortran for IBM 7090, can be run for grids not larger than 50 x 150. The number of grid intervals corresponding to the geometric parameters shown in Fig. 1 can be assigned any non-zero, positive integer value compatible with the above limits. The hydrostatic bearing may have two or three pairs of wells. Any one of the wells may be specified to have unit pressure. Iteration with single steps may or may not be requested. When iteration is requested, the number of iterations must be specified.

The input of the program should be compatible with the Fortran statements:

```
   READ INPUT TAPE 5, 1, IA, IB, IC, ID, IE, IF,
   IM, IPW, ITI, ITO, ITS, AL
   READ INPUT TAPE 5, 11, KMAX, IPUNCH,
   KM, PL
1  FORMAT (11I6, F6.3)
11 FORMAT (3I6, F15.5)
```

where the symbols are explained in Table 1. The listing of the Fortran program is given on the following pages.

Table 1. Explanation of symbols appearing in input statements

IA	Number of grid intervals in a well in y-direction. $a = \text{IA}^*\Delta y$, IA > 1
IB	Number of grid intervals between a well and the nearby outside boundary of pad in y-direction. $b = \text{IB}^*\Delta y$, IB > 1
IC	Number of grid intervals between two neighboring wells in y-direction. $c = \text{IC}^*\Delta y$, IC > 1
ID	Number of grid intervals in a well in x-direction. $d = \text{ID}^*\Delta x$, ID > 1
IE	Number of grid intervals between a well and the nearby outside boundary of pad in x-direction. $e = \text{IE}^*\Delta x$, IE > 1
IF	Number of grid intervals between two neighboring wells in x-direction. $f = \text{IF}^*\Delta x$, IF > 1
IM	Number of well pairs (or number of wells in a row in x-direction). $2 \leq \text{IM} \leq 3$
IPW	The number of the well which has unit pressure. This number should be obtained by counting the wells row-wise. For example, for the pressurized well in Fig. 9 IPW = 3.
ITI	Logical number of a binary tape used by upper-lower procedure.
ITS	Logical number of a binary tape used by upper-lower procedure.
ITO	Logical number of a binary tape used by upper-lower procedure.
KMAX	An indicator for method of solution. If KMAX = 1, the program is requested to do iteration only; otherwise, it will first apply upper-lower procedure.
IPUNCH	An indicator for punched output. If IPUNCH = 1, the program also produces punched output; otherwise, it deletes the punched output.

```
*     LIST8
*     LABEL
C     SOLUTION OF LAPLACIAN FOR HYDROSTATIC BEARING
      DIMENSION A(50,50),B(50,50),C(50),A1(50,50),B1(50,50),C1(50)
     1,P(50,150)
      COMMON IA,IB,IC,ID,IE,IF,IM,IPW,AL,IH,IV,BETA,IH2,IV2,ITI,ITO,ITS,
     1A,B,C,A1,B1,C1,P
      READ INPUT TAPE 5,1,IA,IB,IC,ID,IE,IF,IM,IPW,ITI,ITO,ITS,AL
      READ INPUT TAPE 5,11,KMAX,IPUNCH,  KM,PL
      IH=2*(IA+IB)+IC+1
      IV=2*IE+(IM-1)*IF+IM*ID+1
      FIV=IV-1
      DX=PL/FIV
      DY=DX*AL
      WRITE OUTPUT TAPE 6,2,DX,DY,IA,IB,IC,ID,IE,IF,IM,IPW
      WRITE OUTPUT TAPE 6,12,PL
      IH2=IH-2
      IV2=IV-2
      IF (KMAX-1) 23,7999,23
23    WRITE OUTPUT TAPE 6,7,KM
7     FORMAT ( 36H1THE PROGRAM IS REQUESTED TO PERFORM,I7,3X, 40H ITERAT
     1IONS AFTER UPPER LOWER PROCEDURE.)
      LFF=IE
      LFS=IE+ID+1
      LSF=LFS+IF-1
      LSS=LSF+ID+1
      IF(IM-2)25,25,26
 26   LTF=LSS+IF-1
      LTS=LTF+ID+1
 25   BETA=-2.*(1.+AL*AL)
      REWIND ITI
      REWIND ITO
      REWIND ITS
      L1=IB-1
      L2=IA+IB+1
      L3=IA+IB+IC-1
      L4=IA+IB+IC+IA+1
      DO 30 I=1,IH2
      C(I)=0.
      C1(I)=0.
      DO 29 J=1,IH2
      A(I,J)=0.
      B(I,J)=0.
      A1(I,J)=0.
      B1(I,J)=0.
 29   CONTINUE
 30   CONTINUE
      DO 40 I=1,IH2
      A(I,I)=AL*AL
      B(I,I)=BETA
      IF(I-1)40,35,32
 32   IF (IH2-I) 40,36,34
```

```
 36 B(I,I-1)=1.
    GO TO 40
 35 B(1,I+1)=1.
    GO TO 40
 34 B(I,I-1)=1.
    GO TO 35
 40 CONTINUE
    DO 100 I=1,IV2
    IF (I-LFS) 52,51,51
 52 IF (I-LFF)65,54,54
 51 IF (I-LSF)65,57,57
 57 IF (I-LSS)58,59,59
 59 IF (IM-2)65,65,61
 61 IF (I-LTF)65,63,63
63  IF (I-LTS)64,65,65
 65 WRITE TAPE ITI, ((A(M,N),B(M,N),N=1,IH2),C(M),M=1,IH2)
    GO TO 100
 54 ITCH=1
    GO TO 75
 58 ITCH=2
    GO TO 75
64  ITCH=3
 75 DO 90 K=1,IH2
    IF (K-L1)80,80,77
 77 IF (K-L2)78,79,79
 79 IF (K-L3)80,80,81
81  IF (K-L4) 89,80,80
78  IF (ITCH-IPW) 82,83,82
89  IF (ITCH+IM-IPW) 82,83,82
82  C1(K)=C(K)
    GO TO 84
83  C1(K)=1.
84  A1(K,K)=0.
    B1(K,K)=1.
    GO TO 90
 80 A1(K,K)=A(K,K)
    IF (K-1)85,85,86
 86 IF (IH2-K)100,87,88
 85 B1(1,1)=BETA
    B1(1,2)=1.
    GO TO 90
 88 B1(K,K-1)=B(K,K-1)
    B1(K,K)=B(K,K)
    B1(K,K+1)=B(K,K+1)
    GO TO 90
 87 B1(K,K-1)=B(K,K-1)
    B1(K,K)=BETA
 90 CONTINUE
    WRITE TAPE ITI, ((A1(M,N),B1(M,N),N=1,IH2),C1(M),M=1,IH2)
    DO 95 M=1,IH2
    C1(M)=0.
    DO 94 N=1,IH2
```

```
      A1(M,N)=0.
   94 B1(M,N)=0.
   95 CONTINUE
  100 CONTINUE
      REWIND ITI
      CALL MITSUB
      REWIND ITO
      DO 110 I=1,IH
      DO 110 J=1,IV
110   P(I,J)=0.
      DO 120 J=1,IV2
      JJ=IV2-J+1
      READ TAPE ITO, (P(I+1,JJ+1),I=1,IH2)
  120 CONTINUE
      GO TO 7998
7999  WRITE OUTPUT TAPE 6,8,KM
8     FORMAT ( 36H1THE PROGRAM IS REQUESTED TO PERFORM,I7,3X,17H ITERATI
     1ONS ONLY.)
7998  IF (IPW-IM) 6102,6102,6101
6102  IPC=2*IPW-1
      GO TO 6103
6101  IPC=2*(IPW-IM)
6103  K=0
      JI=IE+1
      JL=JI+ID
6888  II=IB+1
      IL=II+IA
      IF (K+1-IPC) 6000,6999,6000
6000  DO 6001 I=II,IL
      DO 6001 J=JI,JL
6001  P(I,J)=0.
      GO TO 6003
6999  DO 6002 I=II,IL
      DO 6002 J=JI,JL
6002  P(I,J)=1.
6003  K=K+1
      GO TO (6100,6200,6100,6400,6100,6600),K
6100  II=IL+IC
      IL=II+IA
      IF (K+1-IPC) 6000,6999,6000
6200  JI=JL+IF
      JL=JI+ID
      GO TO 6888
6400  IF (IM-2) 6200,6600,6200
6600  ALL=AL*AL
      SAND=2.*(1.+ALL)
      IH1=IH-1
      IV1=IV-1
      IF (KM) 6601,8808,6601
6601  DO 351 K=1,KM
      K=K
      SENSE LIGHT 1
      L=1
```

```
8000 GO TO (8100,8200,8300,8400,8500,8600,8700,350),L
8100 II=2
     IL=IB
     JI=2
     JL=IV1
     GO TO 8900
8200 II=II+IB+IA
     IL=IL+IA+IC
     GO TO 8900
8300 II=II+IC+IA
     IL=IL+IA+IB
     GO TO 8900
8400 JI=2
     JL=IE
     II=2
     IL=IH1
     GO TO 8900
8500 JI=JI+IE+ID
     JL=JL+ID+IF
     GO TO 8900
8600 JI=JI+IF+ID
     JL=JL+ID+IF
     GO TO 8900
8700 JI=JI+IF+ID
     JL=JL+ID+IE
     GO TO 8900
8900 DO 8950 I=II,IL
     DO 8950 J=JI,JL
     P(I,J)=(P(I-1,J)+P(I+1,J)+ALL*(P(I,J-1)+P(I,J+1)))/SAND
8950 CONTINUE
     L=L+1
     IF (L-6) 8806,8805,8806
8805 IF (IM-2) 8806,8804,8806
8804 L=L+1
8806 GO TO 8000
 350 CONTINUE
 351 CONTINUE
8808 PSS=0.
     PXM=0.
     PYM=0.
     DO 150 I=1,IH2
     DO 150 J=1,IV2
     FI=I
     FJ=J
     PSS=PSS+P(I+1,J+1)*AL
     PXM=PXM+P(I+1,J+1)*AL*FI*AL
     PYM=PYM+P(I+1,J+1)*AL*FJ
 150 CONTINUE
     PSS=PSS*DX**2
     PXM=PXM*DX**3
     PYM=PYM*DX**3
     DO 200 K=1,IV,10
```

```
      JB=K
      JE=JB+9
      IF (JE-IV) 190,190,201
  190 WRITE OUTPUT TAPE 6,3,JB,JE,((P(I,J),J=JB,JE),I=1,IH )
  200 CONTINUE
  201 DO 202 K=JB,IV
  202 WRITE OUTPUT TAPE 6,4,K,(P(I,K),I=1,IH )
      IF (IPUNCH - 1) 220,210,220
  210 WRITE OUTPUT TAPE 7,5,IA,IB,IC,ID,IE,IF,IM,IPW,AL,DX,DY,((P(I,J),J
     1=1,IV),I=1,IH)
 220  WRITE OUTPUT TAPE 6,6,PSS,PXM,PYM
      CALL EXIT
    1 FORMAT (11I6,F6.3)
    2 FORMAT ( 98H1SOLUTION OF LAPLACIAN FOR HYDROSTATIC BEARING
     1UTKU-BARONDESS  COMMUNICATIONS RESEARCH  JPL,/////, 3H DX,71X,1H=,
     2F13.6    ,/,3H DY,71X,1H=,F13.6,/,11H WELL WIDTH,63X,1H=,I6,5H  *DY
     3,/,39H BORDER WIDTH IN DIRECTION OF WELL PAIR,35X,1H=,I6,5H  *DY,/
     4,37H DISTANCE BETWEEN THE WELLS IN A PAIR,37X,1H=,I6,5H  *DY,/ ,12
     5H WELL LENGTH,62X,1H=,I6,5H  *DX,/,53H BORDER WIDTH IN DIRECTION P
     6ERPENDICULAR TO WELL PAIR,21X,1H=,I6,5H  *DX,/,61H DISTANCE BETWEE
     7N THE WELLS OF ANY TWO CONSECUTIVE WELL PAIRS,13X,1H=,I6,5H  *DX,/
     8,21H NUMBER OF WELL PAIRS,53X, 1H=,I6,      /,52H NUMBER OF WELL
     9 WITH UNIT PRESSURE, COUNTED ROW-WISE,22X,1H=,I6)
    3 FORMAT ( 24H1PRESSURE MATRIX COLUMNS, I10,5X,7HTHROUGH, I10,/////,
     1(10F12.8))
    4 FORMAT (25H1PRESSURE MATRIX COLUMN  ,I10,/////,(F12.8))
    5 FORMAT (8I6,3F8.5,/,(7F10.8))
    6 FORMAT (16H1TOTAL THRUST IS,F20.6,//,51H STATIC MOMENT ABOUT THE T
     1OP EDGE OF THE BEARING IS,F20.6,//,52H STATIC MOMENT ABOUT THE LEF
     2T EDGE OF THE BEARING IS,F20.6)
   11 FORMAT (3I6,F15.5)
   12 FORMAT (/////,22H THE LENGTH OF THE PAD,F20.3)
      END
```

```
      LABEL
      LIST8
      SUBROUTINE MITSUB
      DIMENSION AP(50,50),AB(50,50),SP(50,50),SB(50,50),YP(50),YB(50),VP
     1(50),VB(50),RHS(50),P(50)
      COMMON IA,IB,IC,ID,IE,IF,IM,IPW,AL,IH,IV,BETA,IH2,IV2,ITI,ITO,ITS,
     1AP,AB,SP,SB,YP,YB,VP,VB,RHS,D,P
      READ TAPE ITI,((AB(I,J),SB(I,J),J=1,IH2),YB(I),I=1,IH2)
      M=1
      CALL MATIS (SB,IH2,RHS,M,D,SP)
      IF (D) 20,1000,20
   20 WRITE TAPE ITS,((SP(I,J),AB(I,J),J=1,IH2),YB(I),I=1,IH2)
      DO 25 I=1,IH2
      DO 25 J=1,IH2
   25 SB(I,J)=SP(I,J)
      IH21=IH2-1
      IV21=IV2-1
      DO 100 L=1,IV21
      READ TAPE ITI,((AP(I,J),SP(I,J),J=1,IH2),YP(I),I=1,IH2)
      DO 30 I=1,IH2
      DO 28 K=1,IH2
      P(K)=0.
      DO 27 J=1,IH2
   27 P(K)=P(K)+SB(I,J)*AP(J,K)
   28 CONTINUE
      DO 29 J=1,IH2
   29 SB(I,J)=P(J)
   30 CONTINUE
      DO 40 I=1,IH2
      DO 38 K=1,IH2
      P(K)=0.
      DO 37 J=1,IH2
   37 P(K)=P(K)+SB(I,J)*AB(J,K)
   38 CONTINUE
      DO 39 J=1,IH2
   39 SP(I,J)=SP(I,J)-P(J)
   40 CONTINUE
      CALL MATIS (SP,IH2,RHS,M,D,AB)
      IF (D) 45,1001,45
   45 DO 50 I=1,IH2
      DY=0.
      DO 49 J=1,IH2
   49 DY=DY+SB(I,J)*YB(J)
   50 YP(I)=YP(I)-DY
      WRITE TAPE ITS,((AB(I,J),AP(I,J),J=1,IH2),YP(I),I=1,IH2)
      DO 70 I=1,IH2
      DO 70 J=1,IH2
   70 SB(I,J)=AB(I,J)
      DO 60 I=1,IH2
      YB(I)=YP(I)
      DO 59 J=1,IH2
```

```
   59 AB(I,J)=AP(I,J)
   60 CONTINUE
  100 CONTINUE
      DO 110 I=1,IH2
      DO 105 K=1,IH2
  105 YP(I)=SB(I,K)*YB(K)
  110 CONTINUE
      WRITE TAPE ITO,(YP(I),I=1,IH2)
      BACKSPACE ITS
      DO 200 L=1,IV21
      BACKSPACE ITS
      READ TAPE ITS,((SB(I,J),AB(I,J),J=1,IH2),YB(I),I=1,IH2)
      DO 210 I=1,IH2
      DY=0.
      DO 209 K=1,IH2
 209  DY=DY+AB(I,K)*YP(K)
  210 YB(I)=YB(I)-DY
      DO 220 I=1,IH2
      YP(I)=0.
      DO 219 K=1,IH2
  219 YP(I)=YP(I)+SB(I,K)*YB(K)
  220 CONTINUE
      WRITE TAPE ITO, (YP(I),I=1,IH2)
      BACKSPACE ITS
  200 CONTINUE
      RETURN
 1000 WRITE OUTPUT TAPE 6,10
      GO TO 2000
 1001 LL=L+1
      WRITE OUTPUT TAPE 6,11,LL
 2000 CALL EXIT
 10   FORMAT (60H1THE FIRST DIAGONAL SUBMATRIX IS SINGULAR. SOLUTION DEL
     1ETED.)
 11   FORMAT (23H1THE DIAGONAL SUBMATRIX,I10,31H IS SINGULAR. SOLUTION D
     1ELETED.)
      END
```

```
*       LABEL
*       LIST8
*       FAP
*                SUBROUTINE MATIS
*                THIS SUBROUTINE IS IDENTICAL WITH MATIV ON JPL LIBRARY
*                TAPE (AUGUST 1962) WHICH IS REASSEMBLED FOR 50X50 ARRAY.
        TTL      MATRIX INVERSION
        LBL      MATIS ,X
        ENTRY    MATIS
*                SUBROUTINE MATIS (A,N,B,M,DETERM,C)
*                THIS SUBROUTINE SAVES MATRTX A
 MATIS  SXD BOY,4
        CLA 1,4
        ADD ONE
        STA Z
        CLA 2,4
        STO ZZ+2
        CLA 3,4
        STO ZZ+3
        CLA 4,4
        STO ZZ+4
        CLA 5,4
        STO ZZ+5
        CLA 6,4
        STO ZZ+1
        ADD ONE
        STA Y
        AXT 2500,4
      Z CLA *,4
      Y STO *,4
        TIX *-2,4,1
     ZZ CALL MATIN
        TSX *
        TSX *
        TSX *
        TSX *
        TSX *
        LXD BOY,4
        TRA     7,4
   ONE  DEC 1
   BOY
        END
```

```
      SUBROUTINE MATIN (A,N,B,M,DETERM)
●     LABEL
*     LIST8
C
C     THIS SUBROUTINE IS IDENTICAL WITH MATINV ON JPL LIBRARY TAPE
C     (AUGUST 1962) WHICH IS RECOMPILED DELETING COMMON STATEMENT AND
C     CHANGING DIMENSION STATEMENT FOR 50X50 ARRAYS.
C     MATRIX INVERSION WITH ACCOMPANYING SOLUTION OF LINEAR EQUATIONS
C
      DIMENSION IPIVOT(50),A(50,50),B(50,1),INDEX(50,2),PIVOT(50)
      EQUIVALENCE (IROW,JROW), (ICOLUM,JCOLUM), (AMAX, T, SWAP)
C
C     INITIALIZATION
C
   10 DETERM=1.0
   15 DO 20 J=1,N
   20 IPIVOT(J)=0
   30 DO 550 I=1,N
C
C     SEARCH FOR PIVOT ELEMENT
C
   40 AMAX=0.0
   45 DO 105 J=1,N
   50 IF (IPIVOT(J)-1) 60, 105, 60
   60 DO 100 K=1,N
   70 IF (IPIVOT(K)-1) 80, 100, 740
   80 IF (ABSF(AMAX)-ABSF(A(J,K))) 85, 100, 100
   85 IROW=J
   90 ICOLUM=K
   95 AMAX=A(J,K)
  100 CONTINUE
  105 CONTINUE
  110 IPIVOT(ICOLUM)=IPIVOT(ICOLUM)+1
C
C     INTERCHANGE ROWS TO PUT PIVOT ELEMENT ON DIAGONAL
C
  130 IF (IROW-ICOLUM) 140, 260, 140
  140 DETERM=-DETERM
  150 DO 200 L=1,N
  160 SWAP=A(IROW,L)
  170 A(IROW,L)=A(ICOLUM,L)
  200 A(ICOLUM,L)=SWAP
  205 IF(M) 260, 260, 210
  210 DO 250 L=1, M
  220 SWAP=B(IROW,L)
  230 B(IROW,L)=B(ICOLUM,L)
  250 B(ICOLUM,L)=SWAP
  260 INDEX(I,1)=IROW
  270 INDEX(I,2)=ICOLUM
  310 PIVOT(I)=A(ICOLUM,ICOLUM)
  320 DETERM=DETERM*PIVOT(I)
```

```
C
C     DIVIDE PIVOT ROW BY PIVOT ELEMENT
C
  330 A(ICOLUM,ICOLUM)=1.0
  340 DO 350 L=1,N
  350 A(ICOLUM,L)=A(ICOLUM,L)/PIVOT(I)
  355 IF(M) 380, 380, 360
  360 DO 370 L=1,M
  370 B(ICOLUM,L)=B(ICOLUM,L)/PIVOT(I)
C
C     REDUCE NON-PIVOT ROWS
C
  380 DO 550 L1=1,N
  390 IF(L1-ICOLUM) 400, 550, 400
  400 T=A(L1,ICOLUM)
  420 A(L1,ICOLUM)=0.0
  430 DO 450 L=1,N
  450 A(L1,L)=A(L1,L)-A(ICOLUM,L)*T
  455 IF(M) 550, 550, 460
  460 DO 500 L=1,M
  500 B(L1,L)=B(L1,L)-B(ICOLUM,L)*T
  550 CONTINUE
C
C     INTERCHANGE COLUMNS
C
  600 DO 710 I=1,N
  610 L=N+1-I
  620 IF (INDEX(L,1)-INDEX(L,2)) 630, 710, 630
  630 JROW=INDEX(L,1)
  640 JCOLUM=INDEX(L,2)
  650 DO 705 K=1,N
  660 SWAP=A(K,JROW)
  670 A(K,JROW)=A(K,JCOLUM)
  700 A(K,JCOLUM)=SWAP
  705 CONTINUE
  710 CONTINUE
  740 RETURN
```